

Amendments to the Claims

1 1. (Currently amended) A computer implemented method of performing a
2 transaction in a database system, comprising:

3 receiving a transaction to be performed, wherein the transaction is
4 processed by a plurality of access modules; and

5 before any directive indicating commencement of an end transaction
6 procedure is broadcast to the access modules, performing a flush of a transaction log
7 from volatile storage to non-volatile storage by each of the access modules before
8 ~~execution of an end transaction procedure begins.~~

1 2. (Currently amended) The method of claim 1, further comprising issuing a
2 request to flush the transaction log with a message sent to each of the access modules for
3 performing a last step of the transaction, the last step performed prior to commencement
4 of the end transaction procedure.

1 3. (Currently amended) The method of claim 2, further comprising
2 performing the flush of the transaction log in a data access step prior to commencement
3 of the end transaction procedure to avoid performance of a transaction log flush in the
4 end transaction procedure.

1 4. (Previously presented) The method of claim 2, further comprising
2 determining that the last step is being performed by all of the plurality of access modules
3 involved in the transaction.

1 5. (Original) The method of claim 1, further comprising determining if the
2 transaction log has been flushed before performing the end transaction procedure.

1 6. (Original) The method of claim 5, further comprising avoiding
2 performance of a transaction log flush in the end transaction procedure if the transaction
3 log has been flushed.

- 1 7. (Original) The method of claim 1, further comprising:
2 identifying the transaction as an implicit transaction.
- 1 8. (Currently amended) The method of claim 1, further comprising:
2 performing the end transaction procedure, ~~which follows execution of the~~
3 ~~transaction.~~
- 1 9. (Currently amended) The method of claim 8, performing the end
2 transaction procedure comprising:
3 skipping broadcast of a the directive indicating commencement of the end
4 transaction procedure to the plurality of access modules.
- 1 10. (Currently amended) A computer implemented method of performing an
2 end transaction procedure in a database system, comprising:
3 after commitment of a transaction, a first access module in the database
4 system writing an end transaction indication to a first transaction log portion in volatile
5 storage, the first access module being part of a cluster of access modules; and
6 the first access module sending an end transaction directive to a fallback
7 access module associated with the first access module, the fallback access module being
8 part of the cluster.
- 1 11. (Currently amended) The method of claim 10, wherein the first access
2 module sends the end transaction directive to the fallback access module but not to other
3 access modules in the cluster.
- 1 12. (Original) The method of claim 10, wherein sending the end transaction
2 directive comprises sending an end transaction-part one directive.

1 13. (Currently amended) The method of claim 12, further comprising the first
2 ~~fallback~~ access module broadcasting an end transaction-part two directive to all access
3 modules in the cluster.

1 14. (Currently amended) The method of claim 10, further comprising the
2 fallback access module writing an end transaction indication to a second transaction log
3 portion in volatile storage.

1 15. (Previously presented) The method of claim 10, further comprising the
2 first access module flushing the first transaction log portion from volatile storage to non-
3 volatile storage.

1 16. (Currently amended) The method of claim 10, further comprising the first
2 access module flushing the first transaction log portions from volatile storage to non-
3 volatile storage but the other access modules in the cluster not flushing their respective
4 transaction log portions.

1 17. (Currently amended) A database system comprising:
2 ~~a plurality of storage media, the storage media comprising persistent~~
3 storage;
4 volatile storage; and
5 a plurality of access modules, wherein each access module is coupled to
6 ~~one of the plurality of the persistent storage media and the volatile storage; and~~
7 each of the access modules being adapted to flush a transaction log
8 maintained by the access module from the volatile storage to the persistent storage before
9 ~~execution of any directive indicating commencement of an end transaction procedure~~
10 begins is broadcast to the access modules.

1 18. (Currently amended) The database system of claim 17, further comprising
2 a controller adapted to determine if each access module has flushed the transaction log
3 maintained by the access module before commencement of the end transaction procedure.

1 19. (Currently amended) The database system of claim 18, wherein the
2 controller is adapted to skip sending a directive to perform a transaction log flush if the
3 controller determines that each access module has flushed the transaction log before
4 commencement of the end transaction procedure.

1 20. (Currently amended) The database system of claim 17, further comprising
2 a controller adapted to provide a flush directive with a message to each of the access
3 modules to perform a last step of the transaction before commencement of the end
4 transaction procedure.

1 21. (Currently amended) An article comprising a computer readable storage
2 medium storing instructions for enabling a processor-based system to:
3 receive a transaction to be performed, wherein the transaction is processed
4 by a plurality of access modules;
5 determine that a last step of the transaction involves the plurality of access
6 modules, wherein the last step is performed before ~~execution~~ any directive indicating
7 commencement of an end transaction procedure ~~begins~~ is broadcast to the access
8 modules; and
9 flush a transaction log from volatile storage to a non-volatile storage while
10 the last step is performed by the plurality of access modules.

1 22. (Previously presented) The article of claim 21, further storing instructions
2 for enabling the processor-based system to:
3 perform the end transaction procedure, wherein the end transaction
4 procedure follows execution of the last step of the transaction.

1 23. (Currently amended) The article of claim 22, further storing instructions
2 for enabling a the processor-based system to:
3 avoid broadcast of any directive indicating commencement of the end
4 transaction procedure to the plurality of access modules.

1 24. (Currently amended) A computer implemented method of performing a
2 transaction in a database system, comprising:
3 receiving a transaction to be performed on plural access modules in the
4 database system;
5 maintaining a log in volatile storage to track operations performed in the
6 transaction; and
7 writing the log to persistent storage before ~~start~~ any directive indicating
8 commencement of an end transaction procedure is broadcast to the plural access modules.

1 25. (Original) The method of claim 24, wherein writing the log to persistent
2 storage comprises flushing the log.

1 26. (Original) The method of claim 24, wherein maintaining the log comprises
2 maintaining a transaction log.

1 27. (Original) The method of claim 24, further comprising performing the end
2 transaction procedure, the end transaction procedure comprising writing an end
3 transaction indication into the log.

1 28. (Currently amended) A database system comprising:
2 ~~storage media comprising~~ persistent storage;
3 volatile storage;
4 access modules coupled to the ~~storage media~~ persistent storage and the
5 volatile storage; and
6 a parsing engine coupled to the access modules, the parsing engine
7 adapted to perform one of:
8 (a) providing a directive with a message to perform a last step
9 of a transaction and communicating the directive to the access modules, each access
10 module responsive to the directive to perform a transaction log flush from the volatile

11 storage to the persistent storage before ~~execution~~ any directive indicating commencement
12 of an end transaction procedure ~~begins~~ is broadcast to the access modules; and

13 (b) determining if each of the access modules has performed a
14 transaction log flush before start of the end transaction procedure;

15 the parsing engine adapted to avoid sending a broadcast directive to the
16 access modules to cause performance of a transaction log flush during the end transaction
17 procedure.

1 29. (Previously presented) The method of claim 1, wherein the transaction
2 comprises plural steps, the method further comprising:

3 performing the plural steps prior to performing the end transaction
4 procedure, and

5 wherein performing the flush of the transaction log comprises performing
6 the flush of the transaction log in one of the plural steps.

1 30. (Previously presented) The method of claim 29, wherein performing the
2 plural steps comprises performing, in each of the plural steps, access of relational table
3 data stored in the database system.

1 31. (Currently amended) The method of claim ~~30~~29, wherein performing the
2 flush of the transaction log in one of the plural steps comprises performing the flush of
3 the transaction log in a last one of the plural steps.

1 32. (Currently amended) The method of claim 31, further comprising each
2 access module adding a first entry to the transaction log to redo the transaction by the
3 access module in case of system failure.

1 33. (Currently amended) The method of claim 4, wherein performing the flush
2 of the transaction log is prior to commencement of the end transaction procedure if the
3 last step is performed by all of the plurality of access modules, the method further
4 comprising:

5 performing the flush of the transaction log in the end transaction
6 procedure if the last step is not performed by all of the plurality of access modules.

1 34. (Currently amended) The database system of claim 17, wherein the access
2 modules are further adapted to perform a transaction comprising plural steps, ~~one or more~~
3 ~~of the access modules adapted to perform the plural steps prior to the end transaction~~
4 ~~procedure~~, and the access modules adapted to perform the flush of the transaction log in
5 one of the plural steps.

1 35. (Previously presented) The database system of claim 34, wherein the one
2 of the plural steps comprises a last one of the steps.

1 36. (Previously presented) The database system of claim 35, wherein the
2 transaction log comprises a first entry associated with each access module to enable a
3 redo of the transaction in case of system failure.

1 37. (Previously presented) The database system of claim 36, wherein the
2 transaction log further comprises a second entry associated with each access module to
3 enable an undo of the transaction.

1 38. (Currently amended) The database system of claim 34, further comprising
2 a controller adapted to determine whether a last one of the steps involves all the access
3 modules, and in response to determining that the last one of the steps involves all the
4 access modules, the controller further adapted to send a directive to all the access
5 modules to perform the flush of the transaction log in the last one of the steps.

1 39. (Currently amended) The database system of claim 38, in response to
2 determining that the last step does not involve all access modules, the controller further
3 adapted to send a directive to perform the flush of the transaction log in the end
4 transaction procedure.

1 40. (Currently amended) The article of claim 21, wherein the transaction
2 comprises plural steps, the article further storing instructions for enabling a the processor-
3 based system to:

4 perform the plural steps prior to ~~performing~~ commencement of the end
5 transaction procedure, and

6 wherein performing the flush of the transaction log comprises performing
7 the flush of the transaction log in one of the plural steps.

1 41. (Currently amended) The article of claim 40, wherein performing the
2 plural steps comprises performing, in each of the plural steps, access of relational table
3 data stored in ~~the~~ a database system.

1 42. (Currently amended) The article of claim ~~41~~0, wherein performing the
2 flush of the transaction log in one of the plural steps comprises performing the flush of
3 the transaction log in a last one of the plural steps.

1 43. (Currently amended) The article of claim 42, further storing instructions
2 for enabling a the processor-based system to cause each access module to add a first entry
3 to the transaction log to redo the transaction by the access module in case of system
4 failure.